

Software Development for Mobile Computing, the Internet of Things and Wearable Devices: Inspecting the Past to Understand the Future

Tor-Morten Grønli
Kristiania University College,
Dep. Technology, Oslo, Norway
tor-morten.gronli@kristiania.no

Andreas Biørn-Hansen
Kristiania University College,
Dep. Technology, Oslo, Norway
andreas.biorn-hansen@kristiania.no

Tim A. Majchrzak
University of Agder,
Kristiansand, Norway
timam@uia.no

Abstract

Currently a convergence in software development for mobile computing (including mobile devices and special technology such as wearables) and the Internet of Things (IoT) can be observed. Devices from the fields are becoming part of a joint ecosystem and share the same infrastructure. Moreover, development processes have changed and user requirements have become very heterogeneous. We have been studying this development for a while, also by offering the fitting HICSS minitrack. In this article, we look into the past of software development for mobile devices, in the context of IoT, and for wearables. We analyse joint characteristics and show differences. Then we invite to a discussion that leads to a research outlook. While neither for industry nor for academia the journey is over, the convergence of fields should offer many new possibilities, prevent problems we faced in the past, but also introduce novel challenges.

1. Introduction

Mobile computing, Internet of Things (IoT) and wearable devices are joining forces and becoming part of the same ecosystem and infrastructure. In modern applications of today the focus is (and *should*) be on solving a specific user's or group of users' needs. This has traditionally been done through the investigation of requirements, prototyping, design, implementation and testing [1]. Multi-disciplinary teams collaborating in order to create a satisfactory solution.

Today this has changed. We still need to solve specific needs, but the user environment has become even more heterogeneous, as we continue to investigate throughout this paper. The focus has shifted to not (only) solving needs, but to completing *tasks*. The user is agnostic to what kind of device the task is solved on, and would often utilise more than one device before completion. Further to this, interaction is based on convenience and availability. For instance, a user starts

texting a friend on the phone, and over time as the conversation is flowing s/he is on the move, and begin to consume the messages on a wearable device such as a smartwatch whilst moving. Back at the office s/he will move over to use the stationary computer as input device. Extending further, voice recognition can be enabled by the IoT device residing on the office desk and complete the multi-dimensional interaction.

Mobile technology and IoT form a multidisciplinary field where sensing devices with communication capabilities that can be applied to ordinary objects in order to make the object "smart". They add a whole new dimension of content to the current Internet paradigm [2]. The topic of wearable computing and smart garments have been researched and developed for several decades [3], but only in recent years has become a popular research field. Gartner first introduced *Wearable User Interfaces* into their *Hype Cycle for Emerging Technologies* in 2013, two years after introducing Internet of Things [4]. This exemplifies how a field with root back decades has been re-established with a contemporary consumer focus and created a new arena of possible outlets for software developers.

Because of this extensive heterogeneous environment, it is important that research focuses on established architectures and integration with existing business practices, and not only provides prototypes and single-case studies. The landscape for software development has become highly dynamic and given the diversity of the field, we showcase how mobile devices, Internet of Things and wearables are all parts coming together in one ecosystem. We explore this phenomena by looking into the past, and highlight research contributions and challenges.

Our article makes two contributions. First, we draw a comprehensive picture of software development for mobile computing – both looking at mobile devices in general and at wearables in particular – and IoT. Second, we provide an outlook into the expected research activities and development in the field.

The remainder is structured as follows. First

we present an overview of state-of-the-art in mobile computing, Internet of Things and wearable computing. Following this we venture into the development ecosystems for these three concepts before discussing the current state of affairs. We then present a synthesis in the form of a research outlook.

2. Background

We will in the following section highlight related work, and give an overview of the major research streams and contributions in the fields of mobile computing, Internet of Things and wearable devices.

2.1. Basics

As depicted in 1, the three fields are closely related. Mobile computing spawned the dawn of Internet of Things for the masses. It is no longer confined to industry projects. Building closely on this, the wearable computing paradigm evolved out of sensors becoming common place and its use being proven to a large portion of the general public.

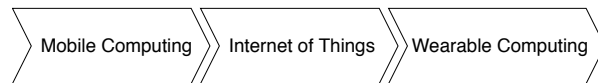


Figure 1. Technological progress

Today, we take all three paradigms for granted and as part of our environment at home, at the office or on the move. To get a better understanding of the way forward until today, we will start looking into the research stream of mobile computing.

2.2. Mobile Computing

Since the release of app-enabled smartphones, counting from the 2007 announcement of iPhone [5], millions of apps have found their way to the various app marketplaces [6]. With the app economy generating billions of USD yearly [7], developing software targeting mobile devices can indeed be a profitable endeavour. We find that numerous studies investigate the “*hows*” of software application development targeting mobile devices.

Within this literature body, we find topics ranging from business predictability and development processes (e.g. [8, 9, 10, 11]) to security (e.g. [12, 13, 14]), to empirical performance-oriented studies (e.g. [15, 16, 17, 18, 19]), and studies focused on end-users’ perception of the quality of mobile apps (e.g. [20, 21, 22, 23, 24, 25, 26]). Research on software development for mobile computing conducted in the intersection of business and software engineering often relies on requirements

or criteria originating from the industry (e.g. [27]), although we rarely encounter case studies in which the *engineering* itself is in focus. This section will introduce research in some frequently encountered topics.

While there exist numerous papers on these and related topics (e.g. [8, 9, 15, 16, 26]), we frequently find ourselves questioning what to focus on going forward. In studies on mobile security, we tend to encounter discussions on attack surfaces (e.g. [28, 29, 30]), ecosystem vulnerabilities (e.g. [31]), and investigations of app-level security measures through permissions (e.g. [32, 33]). However, from the literature, we note that security is not necessarily a topic that practitioners are devoted to, according to interviews conducted by Ahmad *et al.* [34], whom also mention a lack of security-oriented studies in the related research community.

We find that on the topic of performance research, studies span from the introduction of modern smartphone devices, and continue to regularly emerge in academic outlets [15, 16, 35]. Nevertheless, due to the rapid and constantly evolving nature of both software, hardware, and surrounding ecosystems, the continuation of empirical performance measurements is key to keeping up with- and staying relevant to industry and stakeholders. Lastly, studies involving end-users are less frequently encountered than the previous topics traversed. This is especially true in research on cross-platform mobile software development, in which case we only identify a mere handful of user-oriented studies (e.g. [26, 25, 21]). We are interested in seeing this corpus of research extended, as the perception and experience of app usability and quality should be core topics of interest for academics and practitioners alike.

From both the research literature and practitioner outlets we note that traditional approaches and techniques for development of mobile apps are challenged by a wide array of new technologies and novel concepts, some of which we will elaborate on and discuss throughout this paper. While the field of mobile computing goes beyond the development of software applications, the pace at which tools and frameworks facilitating the development of mobile apps are released, is seemingly more rapid than ever. The high turnover of tools could although have serious implications on decision-making, and consequently add quickly to technical debt.

2.3. Internet of Things

The Internet of Things (IoT) are amongst the fastest growing of emerging technologies of today. The IoT technologies and applications are still in their infancy,

and so the academic community must thoroughly address the area [36]. Although IoT was initially meant to describe a network of RadioFrequency ID-enabled devices, it has since been expanded a wider definition of the term incorporating sensors, actuators and agents. According to Perera *et al.* [2], most of the research done on IoT during the last two decades has been focused around prototypes, systems and solutions with a limited number of data sources. It becomes clear that IoT indeed encompasses all devices with a sensor, but there is also a second implication: the huge number of data points that will inevitably be collected is of no use to anyone unless it is processed – and that usually in real time or close to it. The definition also presents us with several implicit challenges, backed by Xu *et al.* [36] and Palattella *et al.* [37]. These include, but are not limited to, privacy, distribution and maintenance, and security concerns in the distributed system that is the IoT.

However, as the technology develops, a need to be context-aware (cf. again the convergence with mobile device challenges [38]) and able to utilise a large number of sensors arises, and with this, the need to develop solutions which implement a strong core architecture, and is flexible and modular enough to be combined with other IoT solutions. From this emerges the paradigms of *edge-computing* or *fog computing*, both used interchangeably and further addressed as fog computing [39]. The purpose of this approach is to take advantage of the powers in low computing devices whereas moving computational resources closer to the edges of the physical application architecture.

2.4. Wearable Devices

Lately, wearables have undergone thorough academic assessment, and enjoyed the attention of numerous researchers and research projects, as displayed through an extensive literature survey by Shrestha and Saxena [40].

While wearables are indeed subject to scrutiny from software engineering and computer science academics, we note that other research traditions and fields find applications for wearables outside of traditional software engineering as well. Within research on health, wearable technology has been applied to cases such as detection of schizophrenia [41], monitoring of animal health [42], and of course personal fitness and health [43]. In the intersection of personal health, gaming and wearables, we find the exceptionally popular Augmented Reality (*AR*)-based mobile game, *Pokémon Go*. In their study, Althoff *et al.* [44] measured the effect of *Pokémon Go* on the activity level of 32 000 players wearing the Microsoft Band wearable

fitness device. The authors report of an overall increase in activity levels across demographics, regardless of gender, weight, and prior levels of activity. Studies similar in nature, e.g. by Fountaine *et al.* [45] also report of an increase in activity levels for their participants in the context of benefits in individual health related to the use and adoption of *Pokémon Go*.

However, while there are definite benefits to the widespread use and adoption of wearables and fitness trackers, numerous ethical and technical issues arise [46]. In a recent study by Classen *et al.* [47], extensive work was conducted in order to uncover vulnerabilities with the Fitbit fitness tracker ecosystem. The team of researchers found that streams of unencrypted fitness data could be captured through a reverse engineering effort of both hardware (wearables) and software (Fitbit app). The implications of such findings can indeed be monumental, especially as Fitbit is one of the leading supplier of personal fitness tracking devices [48]. This is especially relevant as we make note of studies and news articles regarding the use of wearables in commercial situations, e.g. to lower insurance costs (e.g. [49, 50]), and that young insurance holders are positive towards this trend [51].

Another recent event, although involving Strava rather than Fitbit, was the heat mapping of open Strava fitness data to assist in the uncovering and locating of military bases, a work presented on Twitter by Schneider in early 2018 [52]. Not only may this have implications and severe consequences for individuals, it also has the potential of uncovering secret military bases through the use of open fitness data. One could presume that Schneider helped put the governing and use of such data and the associated wearables and fitness mobile apps on the agenda. Indeed, security and safety are kernel to the proposed topics for further research by Shrestha and Saxena [40] in their comprehensive literature survey on wearable computing.

3. Developing for Mobile Devices

A multitude of tools, technologies, frameworks, and boilerplates have been released and made available only in the last year. In the pool of novel and newly released technologies, we find two official tools from Google, including Android Jetpack and Flutter. The former, Jetpack, was announced during the 2018 Google I/O developer conference, and is a set of libraries and best practices for developing native Android apps for mobile, cars, TVs, and smartwatches, aiming to pave a clearer path for architecting and building apps [53]. The latter, Flutter, is a new take on cross-platform app development, with build-targets including iOS and

Android at the time of writing [54], with the possibility of supporting additional platforms, e.g. desktop apps [55]. Flutter builds on ideas implemented in numerous existing cross-platform frameworks, for example the extensible nature of accessing native platform and device features in Cordova and React Native, while simultaneously working to ensure fluid and responsive user interfaces and great performance, according to the development team [54]. In fact, not only will Flutter provide cross-platform apps for Android and iOS, Google's upcoming operating system *Fuchsia* is partially written in Flutter and is reportedly built ground-up to run Flutter-based apps [56]. To the best of our knowledge, neither Flutter nor Jetpack has been subject to academic scrutiny, and should as such be assessed in future work.¹

Within cross-platform mobile development, and the intersection of mobile apps and the web, we find four noteworthy and novel technologies we may encounter more frequently in both research and industry outlets going forward. We briefly introduce and discuss each of these in the coming paragraphs, emphasising their importance in the fast-paced field of practice and research that is mobile computing and its intersection with the field of software engineering.

First, Progressive Web Apps (PWA), a novel approach to extending the possibilities of websites towards providing the same user experience as regular mobile apps. These enhancements consist of on-device installation, offline functionality, background data synchronisation, and handling of push notifications, functionality common to most apps found in the app marketplaces. There have previously been some, although limited, academic scrutiny of Progressive Web Apps, including Malavolta *et al.*'s experiment on the energy efficiency of Service Workers [58], an underlying enabler for Progressive Web Apps. We also find studies by Biørn-Hansen *et al.* [18] and Majchrzak *et al.* [59], the former study providing an overview of concepts and technical possibilities of PWAs in relation to native and cross-platform mobile development, the latter study a comprehensive overview of PWA characteristics, how it builds on concepts of cross-platform mobile development, and suggestions for further research based on previous work.

Going forward, WebAssembly is likely to receive additional attention of industry and academia. Although primarily focusing on enhancing the performance and possibilities of the web, we look at WebAssembly as a technology aiming to enable performance-critical software applications to be deployed to- and made use of inside a web browser, regardless of the form factors

used by end-users. From a technical and historical perspective, JavaScript has been *the* language for front-end web development [60]. Thus, any interactivity a user may have with a website, and any business logic or processing conducted by the front-end of the website, is being controlled by JavaScript. However, WebAssembly allows for the use of other languages such as C, Rust, and C++ in the development of front-end business logic [60, 61]. This is achieved by targeting WebAssembly as the build target of the aforementioned languages, after which the compiled output, a `.wasm` file, can be loaded through JavaScript and executed safely within a web browser [60]. Early research [62, 60] displays promising advantages in terms of performance by having WebAssembly-compiled programs execute and perform tasks found to perform suboptimal in JavaScript, e.g. 3D games and similar CPU intensive software.

Continuing in the context of mobile, encouraging and novel technologies have recently surfaced, including those of React Native DOM and Kotlin/Native. The former, React Native DOM, aims to port React Native to the web, enabling simultaneous cross-platform development of apps that can run not only on mobile platforms (iOS, Android, and Windows Phone), but also have the web as a build target [63]. WebAssembly is the enabling technology rendering it possible to output React Native-based apps to the web browser. The latter, Kotlin/Native, is JetBrains' attempt to run Kotlin-based software outside of the JVM by targeting platforms such as iOS, WebAssembly and Android, effectively allowing cross-platform development in which the end-result apps are native. As Kotlin/Native is available only as a pre-production ready build, we anticipate that future work in our field of research will thoroughly assess the technology upon full release.

When discussing alternative approaches to mobile app development, the *hybrid* development approach is seemingly one of the most common option available for the creation of apps, based on data from app store analyses [64]. Since the advent of Cordova, web-based hybrid apps have been popularised through frameworks including Ionic Framework, PhoneGap, Onsen UI, and Intel XDK. Through Cordova, web developers could interact with native device and platform features through JavaScript-based APIs, allowing non-mobile developers to develop mobile apps using only traditional web development knowledge. However, based on recent surveys including StateOfJS [65] and Stack Overflow's 2018 developer survey [66], and a study by Ahmad *et al.* [34], practitioners' perception of Cordova is at best varying, leaning towards the negative. In an attempt to address issues raised by practitioners, the

¹ Such analyses were driven by our HICSS minitrack, cf. e.g. [57].

Ionic Framework team has only recently launched their own Cordova alternative, *Capacitor*. The goal of Capacitor is to ease the development of custom and third-party modules providing access to native device- and platform features [67], and to provide a unified set of APIs for developing hybrid mobile apps, web-based Progressive Web Apps, and Electron-based desktop apps [67]. Going forward, it will be interesting to see if Capacitor has a positive impact on developers' perception of hybrid mobile apps, which for now is based what Cordova has to offer.

The previously discussed technologies *may* really make a difference in mobile app development when combined. What about WebAssembly-compiled Progressive React Native DOM Web Apps? Or Kotlin/Native-based Progressive Web Apps wherein business logic is as well performing as native mobile apps due to being written in Kotlin and compiled through WebAssembly for web browser execution? A study by Bandi and Fellah [68] investigate design and development-related issues emerging when converting Websites to mobile apps. They highlight numerous issues and challenges in regards to differences in functionality, interaction and design across regular Websites, mobile-friendly websites, and mobile apps.

Taking advantage of the native-like performance of WebAssembly, the app-like experiences provided by Progressive Web Apps, and the novelty of recent frameworks including React Native DOM and Kotlin/Native, the challenges discussed in [68] by Bandi and Fellah may require a revisit near future. Academia focusing on these and similar WebAssembly-related software engineering challenges are some of what we may see addressed in the coming years.

4. Developing for IoT

The Internet of Things is a multidisciplinary field where sensing devices with communication capabilities can be applied to ordinary objects in order to make the object *smart*. As a result, IoT adds a whole new dimension of content to the current Internet paradigm [2]. The primary goal for is to create a *better* world environment, where objects and “things” know exactly what we want and need, and then act towards that goal, without explicit instructions. Development for IoT has grown out of the mobile ecosystem and established itself as a separate field for industry, academics, and businesses. The amount of money being invested in the domain is growing rapidly, although there are disparities in hobby- and consumer-grade equipment, e.g. the existence of \$3 USD WiFi-enabled microchips (ESP8266) allowing for the development

of Internet-connected *things*, and production-grade equipment providing hardware reliability. Within networking, we find novel technologies including 5G and narrowband IoT, and communication infrastructure embedded in more and more everyday devices, leading to a rapidly expanding ecosystem. At the intersection of the technology and *things*, we find the possibilities and constraints of Internet-connected devices with finite energy sources versus the introduction of 5G networks and narrowband technology allowing for extended life of battery-powered systems.

There are several approaches to solving the issues of connectivity within IoT, among them are wireless mesh networks in which several devices connect to the Internet through each other without the need for the Internet access point to have range to every device [69]. Alternatives range to standard WiFi endpoints, if cost allows, to leveraging newer technology like 5G [70]. Wireless mesh networks incur relatively low costs and relatively poor performance due to its distributed and “chained” nature. In addition, nodes much stay within a certain distance of each other. Even if the range is greater, it is still not greater than the broadcast range of a single device. While 5G supports high speeds and greater geographical distribution it comes at an increased cost [70], and has a higher threat surface as every device is generally connected directly to the Internet, where as in a mesh network you would typically have a single gateway.

Security considerations within IoT is a balance between cost and sufficient implementation [71]. Looking at communication between devices and attacks, IoT functions much like any other architecture. Communication can be secured through encryption; input can be screened; software engineering security techniques can be applied in the application code.

IoT development usually follows either of two main strategies. One is to use tooling and frameworks from third party vendors such as the IBM Watson IoT and the Bluemix Cloud platform [72]. This gives the advantage of a complete ecosystem with modules, tools, examples and – typically – an active community. The drawback are vendor lock-in to the given APIs and the inability to control future development. On the contrary, open standards can be used, and through layered approaches traditional web tools and frameworks such as JSON, REST and HTTP based communication can be applied down to the sensor level [73].

5. Developing for Wearables

Wearable devices are emerging as an extension of the well-established smartphone. Now spanning across

a large number of categories (see also [74]), they increasingly gain market penetration. The primary use still resides in consumer electronics as most wearables pose features for enhancing the lifestyle. This typically includes activity tracking, e.g. of steps taken, notifying about time to move, and monitoring health related data such as weight and height, and heart rate. Typical devices include Fitbit, Jawbone and similar wristbands.

Moreover, so called smart watches have spread in the consumer segment. These include Apple iWatch and Android watches e.g. from Samsung, Fossil and LG. Moreover, app-enabled sport watches are offered from Polar, Garmin, TomTom and others. Fragmentation of devices, platforms, and operating systems is common [74]. There is frequent use of vendor based cloud solutions, proprietary development languages and standardised heterogeneous software, tools and frameworks.

An example of a wearables running a proprietary platform is Fitbit. The apps developed for the Fitbit OS will only run on a subset of the Fitbit wearable devices, namely Ionic and Versa as of June 2018 [75]. This excludes most of the Fitbit wearables, e.g. Flex, Alta, and Ace. Thus, if a developer is interested in extracting data from a Fitbit user's device, the fragmentation renders development complex due to the heterogeneous nature of the product-line. While there are challenges to be addressed when developing for wearables within the same product-lines, developing software across multiple heterogeneous wearable platforms, e.g. Fitbit, Garmin and Polar, will likely prove even more difficult.

In cross-platform mobile development, simultaneously developing for multiple heterogeneous mobile environments is at the core of the development philosophy [76]. To the best of our knowledge, cross-platform wearable development has not reached the same adoption as in the context of mobile software development. However, during the WWDC 2018 conference, Apple showcased the fifth version of *watchOS*, their wearables-focused operating system driving their Apple Watch product-line. One of the updates found in *watchOS 5* is the system's ability to open and display websites directly on the wearable device. As information regarding WebKit on *watchOS* is limited at the time of writing, further research should revisit the possibilities offered in terms of executing web-based apps directly on the watch.

6. Discussion

Going forward, we *will* see changes in how development of software applications for mobile devices is conducted. Technology companies including

Google and Facebook advocate their own sets of technologies and technical frameworks for application development, including Flutter, Android JetPack, and React Native. Simultaneously, open web standards including WebAssembly and Progressive Web Apps emerge, aiming to close the current gap in device performance and user experience between the web and what is provided by native mobile- and traditional desktop applications.

6.1. What to Expect: a Research Outlook

With Table 1, we aim to provide a general overview of platforms, technical frameworks, and challenges within the overarching fields of Mobile Computing, Internet of Things, and Wearable Computing. Thus, our intent is to provide concrete ideas as to which novel frameworks and platforms can be used in the assessment of the challenges we find. It is important to note that the content of Table 1 is not exhaustive; it does not list *all* platforms, frameworks and challenges. We rather focus on what we deem to be of interest to both academia and the industry based on our assessment of the literature and communities surrounding the three overarching fields of research.

We have look at what *was* and what *is*. Therefore, we now touch upon what might be. There is a divergence between the current situation and the outlook. Moreover, the current reality of software development for mobile devices, wearables and IoT has shortcomings. Therefore, we propose steps towards a research agenda to close the gaps, to enrich the theory, and to advance the field.

We expect that

- Wearables continue to grow, thus follows the need for academic scrutiny;
- App-enablement [74] will continue to spread;
- Computing power and intelligence moves to the edge of the networks to help offload backend clouds;
- People become less dependent on the smartphone alone, and rely more on assistive technologies including smart watches, fitness trackers, smart glasses, AI-driven voice technology, and more;
- It will be considered to create ecosystem platforms for managing the environment of the personal cloud with smartphone, IoT, and wearable devices
- Heterogenous wearable platforms will see unifying technologies for software development, e.g. through allowing hybrid apps;
- Security, adequate privacy, user experience (UX), and personalisation will have major impact on solutions;

Table 1. Platforms, frameworks and challenges in Mobile Computing, IoT, and Wearable Computing.

Technology	Platforms	Frameworks	Challenges
Mobile Computing	Web (HTML5), iOS, Android	Android Jetpack, Capacitor, WebAssembly, Flutter, NativeScript, React Native, Kotlin/Native, Progressive Web Apps	Device and versioning fragmentation; Pace of new technologies and frameworks; Security; End-user perspectives; Variety of form factors; Web Apps versus traditional apps
Internet of Things	Arduino, Spark, Raspberry Pi, NodeMCU, Teensy, BeagleBone, IBM Watson IoT	JerryScript, IoT.js, Windows 10 IoT, Android Things, Qt, Project Things, Espurino	Finite energy sources; Rural low-connectivity areas; Energy efficiency; Software, firmware and hardware security; Hardware reliability; Vendor lock-ins; Networking protocols and data transferability
Wearable Computing	Fitbit, Garmin, Polar, Apple, Android, Xiaomi, Samsung, Huawei, TomTom	Vendor-specific (WatchKit, Gear SDK, Android Wear, etc.)	Software, firmware and hardware security; Data protection and usage regulations; Computational power; Heterogeneous platforms, SDKs, and product-lines; Ethical perspectives; Reliability of trackers and their generated data

- There will be a convergence in development standards such as web tools, frameworks, and languages (JavaScript) and on device software, such as Kotlin;
- Visualisations, in particular Virtual Reality (VR) and Augmented Reality (AR) will spread;
- Universal design has yet to be properly integrated,
- WebAssembly enables native-like performance in the browser for computationally intensive apps;
- Progressive Web Apps will continue to grow in adoption – It, however, remains to be seen whether they will be *the* unifier; and
- the convergence of the ecosystems of modern mobile computing will continue.

Many of these points can be summarized as moving towards Weisers' vision of ubiquitous computing [77].

Besides that, it is exciting to see how interdisciplinarity plays an increasing role despite the very technical nature of the core of many challenges. However, to address users' needs *merely* improving technology is not enough. This can be particularly witnessed in some fields such as healthcare. Mobile computing, IoT and wearables provide fantastic new possibilities, but some of their main shortcomings – such as security and privacy issues – need to be overcome and their domain-specific implications need to be understood. Therefore, while technological work will remain to be the core of the research activities, multi-disciplinary researcher and truly interdisciplinary thinking will need to drive research projects to leverage what the cohesion of the three fields promises.

6.2. Limitations

This research presented in this article has limitations. Quite obviously, we fathom about the future. Since we are scientists and no prophets, this is a speculative endeavour. Although our outlook is careful, and it is based on a study of the past and an assessment of the status quo, the rapid development in the concerned fields might make some of our estimates obsolete. In addition to this, particularly industrial trends could lead to new settings, paradigms, technologies, and demands that hardly can be estimated. In this regard, it also needs to be stressed that this article is an opinion piece – even though it has a solid foundation. It is not based on work with further experts, such as e.g. found in articles based on Delphi studies. Additionally, it has been written in the setting of the HICSS minitrack we run ([78, 79, 80]; see also next subsection). This is a deliberate boundary.

Besides the limitations concerning the outlook presented in this article, the main limitation is its literature work. We do not present a literature study, although the article is heavily driven by literature. It should, thus, not be confused with one.

6.3. Our Contributions

Our own recent work has been tackling some of the topics we examined in this article – both within the scope of the HICSS minitrack (e.g. [57]) as in other outlets (e.g. [18]). One of us (Majchrzak) is also co-chairing the Web Technologies track at

the ACM Symposium of Applied Computing (SAC). HICSS has developed to be close to the information systems (IS) community, both regarding topics and publication (through the Association for Information Systems (AIS)). The topics we propose here falls under the umbrella of software engineering and thereby mark the border between IS and computer science. We think that this greatly drives their appeal!

Our own research will aim for the following goals:

- Further deepen the understanding of cross-platform app development. In particular, we intend to provide recommendations for cross-platform framework choice and to unify the work on cross-platform technologies for different devices;
- Extend our understanding of end-users' perception of quality and user experience of apps built using cross-platform technologies;
- Establish platform and language independent guidelines for evaluation and comparison of mobile application development;
- Partake in the assessment of challenges listed as part of Table 1, all of which would be interesting topics for the HICSS minitrack;
- Scientifically document the work on possibly unifying technologies such as Progressive Web Apps (PWA) and WebAssembly, and the intersection of these;
- Investigate and bridge the gap between lightweight architecture for IoT with mobile / wearable applications and edge computing; and
- Eventually and successively enrich the theory on modern mobile computing including the software development side of the Internet of Things.

For our HICSS minitrack, the journey is far from over. It is needless to say that software development remains a relevant topic; how the future topics of development for mobile devices and the Internet of Things will be named is impossible to anticipate, though. We, thus, intend to keep updating the minitrack's description to reflect upon these developments. Thereby, the minitrack itself should become a *documentation of the progress* in the field.

7. Conclusion

In this article we have discussed software development for mobile devices, the Internet of Things, and wearables. For the three fields we have taken a literature-based and practice-acknowledged look. We took this opportunity while running the alike-themed minitrack under the Software Technology track at the Hawaii International Conference on Systems

Science (HICSS). Successively, we have assessed the status quo and taken a peek into the future. We thereby hope to shed some light into what *might* be the next research topics, and what we might see in the HICSS track in the next few years.

We hope that other researchers – and maybe even practitioners – will find this useful and encouraging. Many of the challenges listed in this article likely should be closed soon, and some of the contributions to this may be witnessed as part of HICSS' presentations.

Our own work will be centered on the research outlook that we proposed. In particular, we will be looking at cross-platform technology, document and assess the technological progress, and amend the theory on modern mobile computing.

References

- [1] T. R. Bhongale, S. S. Dhamnekar, A. K. Sanadi, S. K. Nandgave, A. A. Pawar, and S. A. Pardeshi, "Survey on tools and technologies applicable for mobile application development," *IRJET*, vol. 5, 2018.
- [2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3] S. Mann, "Smart clothing: The shift to wearable computing," *CACM*, vol. 39, no. 8, pp. 23–24, 1996.
- [4] Gartner Group, "Gartner group internet of things." <https://www.gartner.com/technology/research/internet-of-things/>, 2018.
- [5] M. Macedonia, "iPhones Target the Tech Elite," *Computer*, vol. 40, pp. 94–95, 2007.
- [6] D. Loesche, "Infographic: The biggest app stores." <https://www.statista.com/chart/12455/number-of-apps-available-in-leading-app-stores/>, 2018.
- [7] Statista, "Mobile app revenues 2015-2020." <https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/>, 2016.
- [8] T. A. Majchrzak, J. Ernstring, and H. Kuchen, "Achieving business practicability of model-driven cross-platform apps," *OJIS*, vol. 2, no. 2, pp. 3–14, 2015.
- [9] C. Rieger and H. Kuchen, "A process-oriented modeling approach for graphical development of mobile business apps," *Computer Languages, Systems & Structures*, vol. 53, pp. 43–58, 2018.
- [10] J. Ernstring, C. Rieger, F. Wrede, and T. A. Majchrzak, "Refining a reference architecture for Model-Driven business apps," in *12th WEBIST*, pp. 307–316, Scitepress, 2016.
- [11] T.-M. Grønli and G. Ghinea, "Meeting quality standards for mobile application development in businesses: A framework for Cross-Platform testing," in *49th HICSS*, pp. 5711–5720, 2016.
- [12] K. Choo, L. Rokach, C. Bettini, *et al.*, "Mobile security and privacy: advances, challenges and future research directions," *Pervasive and Mobile Computing*, vol. 32, pp. 1–112, 2016.

- [13] A.-R. Sadeghi, "Mobile security and privacy: The quest for the mighty access control," in *Proc. 18th ACM SACMAT*, pp. 1–2, ACM, 2013.
- [14] Q. Li and G. Clark, "Mobile security: A look ahead," *IEEE Security Privacy*, vol. 11, pp. 78–81, Jan 2013.
- [15] M. Ciman and O. Gaggi, "An empirical analysis of energy consumption of cross-platform frameworks for mobile development," *Pervasive and Mobile Computing*, 2016.
- [16] Y. Ma, X. Liu, Y. Liu, Y. Liu, and G. Huang, "A tale of two fashions: An empirical study on the performance of native apps and web apps on android," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 990–1003, 2018.
- [17] L. Corral, A. Sillitti, and G. Succi, "Mobile multiplatform development: An experiment for performance analysis," *Procedia Comput. Sci.*, vol. 10, pp. 736–743, 2012.
- [18] A. Bjørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, "Progressive web apps for the unified development of mobile applications," in *Web Information Systems and Technologies*, vol. 322 of *LNCSE*, Springer, 2018. (to appear).
- [19] M. Willocx, J. Vossaert, and V. Naessens, "Comparing performance parameters of mobile app development strategies," in *IEEE/ACM MOBILESoft*, pp. 38–47, IEEE, 2016.
- [20] G. Huang, M. Xu, F. Xiaozhu Lin, Y. Liu, Y. Ma, S. Pushp, and X. Liu, "ShuffleDog: Characterizing and adapting User-Perceived latency of android apps," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2913–2926, 2017.
- [21] E. Noei, M. D. Syer, Y. Zou, A. E. Hassan, and I. Keivanloo, "A study of the relation of mobile device attributes with the user-perceived quality of android apps," *Empirical Software Engineering*, vol. 22, no. 6, pp. 3088–3116, 2017.
- [22] I. Malavolta, S. Ruberto, T. Soru, and V. Terragni, "End users' perception of hybrid mobile apps in the google play store," in *IEEE Int. Conf. on Mobile Services*, pp. 25–32, IEEE, 2015.
- [23] J. Huhtala, A.-H. Sarjanoja, J. Mäntyjärvi, M. Isomursu, and J. Häkkinen, "Animated UI transitions and perception of time: A user study on animated effects on a mobile screen," in *Proc. 28th SIGCHI*, pp. 1339–1342, ACM, 2010.
- [24] I. T. Mercado, N. Munaiah, and A. Meneely, "The impact of cross-platform development approaches for mobile applications from the user's perspective," in *Proc. WAMA*, pp. 43–49, ACM, 2016.
- [25] G. Mesfin, G. Ghinea, D. Midekso, and T.-M. Grønli, "Evaluating usability of Cross-Platform smartphone applications," in *Mobile Web Information Systems*, LNCSE, pp. 248–260, Springer, 2014.
- [26] E. Angulo and X. Ferre, "A case study on Cross-Platform development frameworks for mobile applications and UX," in *Proc. XV Int. Conf. on HCI*, p. 27, ACM, 2014.
- [27] T. A. Majchrzak, A. Bjørn-Hansen, and T.-M. Grønli, "Comprehensive analysis of innovative Cross-Platform app development frameworks," in *Proc. 50th HICSS*, pp. 6162–6171, ScholarSpace, 2017.
- [28] T. Luo, H. Hao, W. Du, Y. Wang, and H. Yin, "Attacks on WebView in the android system," in *Proc. 27th ACSAC*, pp. 343–352, ACM, 2011.
- [29] W. Bao, W. Yao, M. Zong, and D. Wang, "Cross-site scripting attacks on android hybrid applications," in *Proc. 2017 Int. Conf. on Cryptography, Security and Privacy*, pp. 56–61, ACM, 2017.
- [30] M. L. Hale and S. Hanson, "A testbed and process for analyzing attack vectors and vulnerabilities in hybrid mobile apps connected to restful web services," in *IEEE World Congress on Services*, pp. 181–188, 2015.
- [31] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, "Android security: A survey of issues, malware penetration, and defenses," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.
- [32] S. Pooryousef and M. Amini, "Fine-grained access control for hybrid mobile applications in android using restricted paths," in *13th ISCISC*, pp. 85–90, IEEE, 2016.
- [33] P. H. Phung, A. Mohanty, R. Rachapalli, and M. Sridhar, "HybridGuard: A Principal-Based permission and Fine-Grained policy enforcement framework for Web-Based mobile applications," in *IEEE SPW*, pp. 147–156, IEEE, 2017.
- [34] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An empirical study of investigating mobile applications development challenges," *IEEE Access*, vol. 6, pp. 17711–17728, 2018.
- [35] L. Delía, N. Galdamez, L. Corbalan, P. Pesado, and P. Thomas, "Approaches to mobile application development: Comparative performance analysis," in *2017 Computing Conference*, pp. 652–659, 2017.
- [36] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [37] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.
- [38] T. A. Majchrzak and M. Schulte, "Context-dependent testing of applications for mobile devices," *Open Journal of Web Technologies (OJWT)*, vol. 2, no. 1, pp. 27–39, 2015.
- [39] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. 1st MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.
- [40] P. Shrestha and N. Saxena, "An offensive and defensive exposition of wearable computing," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 92:1–92:39, 2017.
- [41] M. Cella, Ł. Okruszek, M. Lawrence, V. Zarlenga, Z. He, and T. Wykes, "Using wearable technology to detect the autonomic signature of illness severity in schizophrenia," *Schizophr. Res.*, vol. 195, pp. 537–542, 2018.
- [42] T. Norton, D. Piette, V. Exadaktylos, and D. Berckmans, "Automated real-time stress monitoring of police horses using wearable technology," *Appl. Anim. Behav. Sci.*, vol. 198, pp. 67–74, 2018.
- [43] S. Y. Lee and K. Lee, "Factors that influence an individual's intention to adopt a wearable healthcare device: The case of a wearable fitness tracker," *Technol. Forecast. Soc. Change*, vol. 129, pp. 154–163, 2018.
- [44] T. Althoff, R. W. White, and E. Horvitz, "Influence of pokémon go on physical activity: Study and implications," *J. Med. Internet Res.*, vol. 18, no. 12, p. e315, 2016.

- [45] C. J. Fountaine, E. J. Springer, and J. R. Sward, "A descriptive study of objectively measured pokémon GO playtime in college students," *Int. J. Exerc. Sci.*, vol. 11, no. 7, pp. 526–532, 2018.
- [46] L. H. Segura Anaya, A. Alsadoon, N. Costadopoulos, and P. W. C. Prasad, "Ethical implications of user perceptions of wearable devices," *Sci. Eng. Ethics*, vol. 24, no. 1, pp. 1–28, 2018.
- [47] J. Classen, D. Wegemer, P. Patras, T. Spink, and M. Hollick, "Anatomy of a vulnerable fitness tracking system: Dissecting the fitbit cloud, app, and firmware," *Proc. ACM IMWUT*, vol. 2, no. 1, pp. 5:1–5:24, 2018.
- [48] Statista, "Wearables shipments worldwide by vendor 2014-2018." <https://www.statista.com/statistics/435933/>, 2018.
- [49] S. Paluch and S. Tuzovic, "Leveraging pushed self-tracking in the health insurance industry: How do individuals perceive smart wearables offered by insurance organization?," in *Proc. 25th ECIS*, pp. 2732–2743, AISel, 2017.
- [50] R.-B. Wiegard and M. H. Breitner, "Smart services in healthcare: A risk-benefit-analysis of pay-as-you-live services from customer perspective in germany," *Electronic Markets*, 2017.
- [51] PricewaterhouseCoopers, "The wearable future," tech. rep., PwC, 2014.
- [52] T. Schneider, "Fitness and social media company strava releases activity heat map. excellent for locating military bases." <https://twitter.com/tobiaschneider/status/957317886112124928>, 2018.
- [53] Android Developers, "Android jetpack." <https://developer.android.com/jetpack/>.
- [54] Flutter Team, "FAQ - flutter." <https://flutter.io/faq/>, 2018.
- [55] S. Morgan and A. Davies, "Flutter desktop embedding." <https://github.com/google/flutter-desktop-embedding>, 2018.
- [56] K. Bradshaw, "How flutter is paving the way for fuchsia." <https://9to5google.com/2018/03/02/fuchsia-friday-first-fuchsia-app/>, 2018.
- [57] T. A. Majchrzak, A. Bjørn-Hansen, and T.-M. Grønli, "Comprehensive analysis of innovative cross-platform app development frameworks," in *Proc. 50th HICSS-50*, AISel, 2017.
- [58] I. Malavolta, G. Procaccianti, P. Noorland, and P. Vukmirović, "Assessing the impact of service workers on the energy efficiency of progressive web apps," in *Proc. 4th MOBILESoft*, pp. 35–45, IEEE Press, 2017.
- [59] T. A. Majchrzak, A. Bjørn-Hansen, and T.-M. Grønli, "Progressive web apps: the definite approach to Cross-Platform development?," in *Proc. 51st HICSS*, pp. 5735–5745, 2018.
- [60] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, and J. F. Bastien, "Bringing the web up to speed with WebAssembly," in *Proc. 38th PLDI*, pp. 185–200, ACM, 2017.
- [61] N. Fitzgerald, A. Arora, M. Gattozzi, A. Turon, P. Hertleif, J. L. Lucchetta, and A. Crichton, "RustWASM." <https://github.com/rustwasm/team>, 2018.
- [62] M. Reiser and L. Bläser, "Accelerate JavaScript applications by cross-compiling to WebAssembly," in *Proc. 9th ACM VMIL*, pp. 10–17, ACM, 2017.
- [63] V. Riemer and M. Zilberman, "react-native-dom." <https://github.com/vincentriemer/react-native-dom>, 2018.
- [64] N. Viennot, E. Garcia, and J. Nieh, "A measurement study of google play," in *ACM SIGMETRICS conference*, vol. 42, pp. 221–233, ACM, 2014.
- [65] S. Greif, R. Benitte, and M. Rambeau, "Mobile & desktop frameworks." <https://stateofjs.com/2017/mobile/results>, 2017.
- [66] Stack Overflow, "Stack overflow developer survey 2018." <https://insights.stackoverflow.com/survey/2018/>, 2018.
- [67] Ionic Framework, "Capacitor: Native progressive web apps." <https://capacitor.ionicframework.com/docs/>, 2018.
- [68] A. Bandi and A. Fellah, "Design issues for converting websites to mobile sites and apps: A case study," in *2017 ICCMC*, pp. 652–656, IEEE, 2017.
- [69] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
- [70] I. Chih-Lin, S. Han, Y. Chen, and G. Li, "Trillions of nodes for 5g!," in *IEEE/CIC Communications in China (ICCC)*, pp. 246–250, IEEE, 2014.
- [71] M. J. Covington and R. Carskadden, "Threat implications of the internet of things," in *Proc. 5th CyCon*, pp. 1–12, IEEE, 2013.
- [72] IBM, "Watson internet of things." <https://www.ibm.com/internet-of-things/>, 2018.
- [73] T.-M. Grønli, G. Ghinea, and M. Younas, "A lightweight architecture for the web-of-things," in *Proc. MobiWIS*, pp. 248–259, Springer, 2013.
- [74] C. Rieger and T. A. Majchrzak, "Conquering the Mobile Device Jungle: Towards a Taxonomy for App-Enabled Devices," in *Proc. 13th WEBIST*, pp. 332–339, SciTePress, 2017.
- [75] Fitbit, "FitbitOS - meet the watch experience designed with your goals in mind." <https://www.fitbit.com/fitbitos>, 2018.
- [76] C. Rieger and T. A. Majchrzak, "Weighted evaluation framework for cross-platform app development approaches," in *LNBIP*, pp. 18–39, Springer, 2016.
- [77] M. Weiser, "The Computer for the 21st Century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999.
- [78] T. A. Majchrzak and T. Grønli, "Introduction to the minitrack on software development for mobile devices, wearables, and the internet-of-things," in *51st HICSS*, AISel, 2018.
- [79] T. A. Majchrzak and T. Grønli, "Introduction to mobile app development minitrack," in *50st HICSS*, AISel, 2017.
- [80] T. A. Majchrzak and H. Heitkötter, "Introduction to the mobile app development minitrack," in *49st HICSS*, p. 5682, IEEE Computer Society, 2016.